Search Results

Search Results for: [reload<AND>(((class loader)))]
Found 17 of 129,763 searched.

Search within Results

1 Dynamic class loading in the Java virtual machine 88%

Sheng Liang , Gilad Bracha

ACM SIGPLAN Notices , Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications October 1998

Volume 33 Issue 10

Class loaders are a powerful mechanism for dynamically loading software components on the Java platform. They are unusual in supporting all of the following features: *laziness, type-safe linkage, user-defined extensibility*, and *multiple communicating namespaces*.We present the notion of class loaders and demonstrate some of their interesting uses. In addition, we discuss how to maintain type safety in the presence of user-defined dynamic class loading.

2 Programming techniques: Manipulation of Java agent bytecode to add roles 83%

Giacomo Cabri , Luca Ferrari , Letizia Leonardi

Proceedings of the 2nd international conference on Principles and practice of programming in Java June 2003

Roles are a powerful paradigm to develop distributed applications based on agents, especially when they are in need of interacting with other entities. An agent-oriented approach requires that roles are conceived as first-class entities, and at the same time that roles are dynamically embedded into agents at runtime. In this paper we propose an approach that addresses such requirements, enabling Java agents to dynamically assume roles. We present a mechanism that modifies the agent bytecode to a ...

3 Technical Correspondence: Unloading Java classes that contain static fields 82%

C. E. McDowell , E. A. Baldwin

ACM SIGPLAN Notices January 1998

Volume 33 Issue 1

In Java, the definition of a "program" is a bit fuzzy. A Java applet is essentially a Java application (i.e. program) that can be executed by a Java enabled Web browser (i.e. an OS). An applet running inside of a browser was intended to be analogous to a conventional application running under on OS, hence the netcentric "browser is your OS" model. However, as currently implemented, this analogy breaks down with regard to the system

resources allocated for classes and in particular for static fie ...

4 A scalable architecture for multi-threaded JAVA applications 80%

M. Mrva , K. Buchenrieder , R. Kress

Proceedings of the conference on Design, automation and test in Europe February 1998

The paper presents a scalable architecture for multi-threaded Java applications. Threads enable modeling of concurrent behavior in a more or less natural way. Thus threads give a migration path to multi-processor machines. The proposed architecture consists of multiple application-specific processing elements, each able to execute a single thread at one time. The architecture is evaluated by implementing a portable and scalable Java machine onto an FPGA board for demonstration.

5 Reflections on remote reflection 77%

Michael Richmond , James Noble

Australian Computer Science Communications , Proceedings of the 24th Australasian conference on Computer science January 2001

Volume 23 Issue 1

The Java programming language provides both reflection and remote method invocation: reflection allows a program to inspect itself and its runtime environment, remote method invocation allows methods to be invoked transparently across a network. Unfortunately, the standard Java implementations of reflection and remote method invocation are incompatible: programmers cannot reflect on a remote application.We describe how Java systems can be extended to support **Remote Reflection** transparentl ...

6 A selective, just-in-time aspect weaver 77%

Yoshiki Sato , Shigeru Chiba , Michiaki Tatsubori

Proceedings of the second international conference on Generative programming and component engineering September 2003

Dynamic AOP (Aspect-Oriented Programming) is receiving growing interests in both the academia and the industry. Since it allows weaving aspects with a program at runtime, it is useful for rapid prototyping and adaptive software. However, the previous implementations of dynamic AOP systems suffered from serious performance penalties. This paper presents our new efficient dynamic AOP system in Java for addressing the underlying problem. This system called Wool is a hybrid of two approaches. When a ...

7 Partial behavioral reflection: spatial and temporal selection of reification 77%

Éric Tanter , Jacques Noyé , Denis Caromel , Pierre Cointe

ACM SIGPLAN Notices , Proceedings of the 18th ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications October 2003

Volume 38 Issue 11

Behavioral reflection is a powerful approach for adapting the behavior of running applications. In this paper we present and motivate *partial behavioral reflection*, an approach to more efficient and flexible behavioral reflection. We expose the *spatial* and *temporal* dimensions of such reflection, and propose a model of partial behavioral reflection based on the notion of *hooksets*. In the context of Java, we describe a reflective architecture offering appropriate interf ...

8 DrJava: a lightweight pedagogic environment for Java 77%

Eric Allen , Robert Cartwright , Brian Stoler

ACM SIGCSE Bulletin , Proceedings of the 33rd SIGCSE technical symposium on Computer science education February 2002

Volume 34 Issue 1

DrJava is a pedagogic programming environment for Java that enables students to focus on

designing programs, rather than learning how to use the environment. The environment provides a simple interface based on a "read-eval-print loop" that enables a programmer to develop, test, and debug Java programs in an interactive, incremental fashion. This paper gives an overview of DrJava including its pedagogic rationale, functionality, and implementation.

9 Proxy compilation of dynamically loaded Java classes with MoJo 77%

Matt Newsome , Des Watson

ACM SIGPLAN Notices , Proceedings of the joint conference on Languages, compilers and tools for embedded systems: software and compilers for embedded systems June 2002

Volume 37 Issue 7

Interest in Java implementations for resource-constrained environments such as embedded systems has been tempered by concerns regarding its efficiency. Current native compilers for Java offer dramatic increases in efficiency, but have poor support for dynamically-loaded classes, which are typically served by slow interpreters or JIT compilers, the code-size of this latter utterly mismatching the resource constraints of the system.After a brief survey of Ahead-of-Time compilers for Java, we prese ...

10 Formalizing the safety of Java, the Java virtual machine, and Java card 77%

Pieter H. Hartel , Luc Moreau

ACM Computing Surveys (CSUR) December 2001

Volume 33 Issue 4

We review the existing literature on Java safety, emphasizing formal approaches, and the impact of Java safety on small footprint devices such as smartcards. The conclusion is that although a lot of good work has been done, a more concerted effort is needed to build a coherent set of machine-readable formal models of the whole of Java and its implementation. This is a formidable task but we believe it is essential to build trust in Java safety, and thence to achieve ITSEC level 6 or Common Crite ...

11 A secure execution framework for Java 77%

Manfred Hauswirth , Clemens Kerer , Roman Kurmanowytsch

Proceedings of the 7th ACM conference on Computer and communications security November 2000

12 A framework for interprocedural optimization in the presence of dynamic class loading 77%

Vugranam C. Sreedhar , Michael Burke , Jong-Deok Choi

ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation May 2000

Volume 35 Issue 5

Dynamic class loading during program execution in the Java Programming Language is an impediment for generating code that is as efficient as code generated using static whole-program analysis and optimization. Whole-program analysis and optimization is possible for languages, such as C++, that do not allow new classes and/or methods to be loaded during program execution. One solution for performing whole-program analysis and avoiding incorrect execution after a new class is loaded is to in ...

13 Back to the basics: a first class chalkboard and more 77%

Ng S. T. Chong , Masao Sakauchi

Proceedings of the 2000 ACM symposium on Applied computing March 2000

14 Using complete system simulation to characterize SPECjvm98 benchmarks 77%

Tao Li , Lizy Kurian John , Vijaykrishnan Narayanan , Anand Sivasubramaniam , Jyotsna Sabarinathan , Anupama Murthy

Proceedings of the 14th international conference on Supercomputing May 2000

Complete system simulation to understand the influence of architecture and operating systems on application execution has been identified to be crucial for systems design. While there have been previous attempts at understanding the architectural impact of Java programs, there has been no prior work investigating the operating system (kernel) activity during their executions. This problem is particularly interesting in the context of Java since it is not only the application that can invoke ...

15 Zones, contracts and absorbing changes: an approach to software evolution 77%

Huw Evans , Peter Dickman

ACM SIGPLAN Notices , Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications October 1999

Volume 34 Issue 10

This paper describes a novel approach to managing the evolution of distributed, persistent systems at run-time. This is achieved by partitioning a system into disjoint zones, each of which can be evolved without affecting code in any other. Contracts are defined between zones, making type-level interdependencies and inter-zone communication explicit. Programmer supplied code is added to the running system, at the boundary between zones, to constrain the sco ...

16 A specification of Java loading and bytecode verification 77%

Allen Goldberg

Proceedings of the 5th ACM conference on Computer and communications security November 1998

17 A geographically distributed framework for embedded system design and validation 77%

Ken Hines , Gaetano Borriello

Proceedings of the 35th annual conference on Design automation conference May 1998

The difficulty of emb edded system co-design is increasing rapidly due to the incr easing complexity of individual parts, the variety of parts available and pr essure to use multiple processors to me et performanc e criteria. V alidation tools should contain sever al features in order to keep up with this trend, including the ability to dynamically change detail levels, built in protection for intellectual property, and supp ort for gradual migr ation of functionalityfrom a simulati ...

Results 1 - 17 of 17 **short listing**

Your search matched 8 of 1022101 documents.
A maximum of 500 results are displayed, 15 to a page, sorted by Relevance in Descending order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

class loader Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine CNF = Conference STD = Standard

---

1 **JMangler - a framework for load-time transformation of Java class files**
Kniesel, G.; Costanza, P.; Austermann, M.;
Source Code Analysis and Manipulation, 2001. Proceedings. First IEEE International Workshop on , 10 Nov. 2001
Pages:98 - 108

[Abstract] [PDF Full-Text (128 KB)] IEEE CNF

---

2 **Secure Java class loading**
Li Gong;
Internet Computing, IEEE , Volume: 2 , Issue: 6 , Nov.-Dec. 1998
Pages:56 - 61

[Abstract] [PDF Full-Text (64 KB)] IEEE JNL

---

3 **Implementing dynamic language features in Java using dynamic code generation**
Breuel, T.M.;
Technology of Object-Oriented Languages and Systems, 2001. TOOLS 39. 39th International Conference and Exhibition on , 29 July-3 Aug. 2001
Pages:143 - 152

[Abstract] [PDF Full-Text (480 KB)] IEEE CNF

---

4 **Development of soccer agents with object migration**
Maeda, K.;
Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on , Volume: 6 , 12-15 Oct. 1999
Pages:750 - 755 vol.6

[Abstract] [PDF Full-Text (336 KB)] IEEE CNF

5 **A real-time Java system on a multithreaded Java microcontroller**
Pfeffer, M.; Uhrig, S.; Ungerer, T.; Brinkschulte, U.;
Object-Oriented Real-Time Distributed Computing, 2002. (ISORC 2002). Proceedings. Fifth IEEE International Symposium on , 29 April-1 May 2002
Pages:34 - 41

[Abstract] [PDF Full-Text (279 KB)] IEEE CNF

6 **Jato: a compact binary file format for Java class**
Sheng-De Wang; Lin, Y.;
Parallel and Distributed Systems, 2001. ICPADS 2001. Proceedings. Eighth International Conference on , 26-29 June 2001
Pages:467 - 474

[Abstract] [PDF Full-Text (556 KB)] IEEE CNF

7 **Using Java to add "stored procedures" to databases**
Ege, R.K.; Rishe, N.; Jingyu Liu; Lebedev, V.;
Technology of Object-Oriented Languages and Systems, 1999. TOOLS 30. Proceedings , 1-5 Aug. 1999
Pages:322 - 331

[Abstract] [PDF Full-Text (192 KB)] IEEE CNF

8 **Design, and implementation of a Java execution environment**
Chen, F.G.; Ting-Wei Hou;
Parallel and Distributed Systems, 1998. Proceedings., 1998 International Conference on , 14-16 Dec. 1998
Pages:686 - 692

[Abstract] [PDF Full-Text (56 KB)] IEEE CNF

Your search matched 8 of 1022101 documents.
A maximum of 500 results are displayed, 15 to a page, sorted by Relevance in Descending order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

class loader Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine CNF = Conference STD = Standard

---

1 **JMangler - a framework for load-time transformation of Java class files**
Kniesel, G.; Costanza, P.; Austermann, M.;
Source Code Analysis and Manipulation, 2001. Proceedings. First IEEE International Workshop on , 10 Nov. 2001
Pages:98 - 108

[Abstract] [PDF Full-Text (128 KB)] IEEE CNF

---

2 **Secure Java class loading**
Li Gong;
Internet Computing, IEEE , Volume: 2 , Issue: 6 , Nov.-Dec. 1998
Pages:56 - 61

[Abstract] [PDF Full-Text (64 KB)] IEEE JNL

---

3 **Implementing dynamic language features in Java using dynamic code generation**
Breuel, T.M.;
Technology of Object-Oriented Languages and Systems, 2001. TOOLS 39. 39th International Conference and Exhibition on , 29 July-3 Aug. 2001
Pages:143 - 152

[Abstract] [PDF Full-Text (480 KB)] IEEE CNF

---

4 **Development of soccer agents with object migration**
Maeda, K.;
Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on , Volume: 6 , 12-15 Oct. 1999
Pages:750 - 755 vol.6

[Abstract] [PDF Full-Text (336 KB)] IEEE CNF

5 **A real-time Java system on a multithreaded Java microcontroller**
Pfeffer, M.; Uhrig, S.; Ungerer, T.; Brinkschulte, U.;
Object-Oriented Real-Time Distributed Computing, 2002. (ISORC 2002). Proceedings. Fifth IEEE International Symposium on , 29 April-1 May 2002
Pages:34 - 41

[Abstract] [PDF Full-Text (279 KB)] IEEE CNF

6 **Jato: a compact binary file format for Java class**
Sheng-De Wang; Lin, Y.;
Parallel and Distributed Systems, 2001. ICPADS 2001. Proceedings. Eighth International Conference on , 26-29 June 2001
Pages:467 - 474

[Abstract] [PDF Full-Text (556 KB)] IEEE CNF

7 **Using Java to add "stored procedures" to databases**
Ege, R.K.; Rishe, N.; Jingyu Liu; Lebedev, V.;
Technology of Object-Oriented Languages and Systems, 1999. TOOLS 30. Proceedings , 1-5 Aug. 1999
Pages:322 - 331

[Abstract] [PDF Full-Text (192 KB)] IEEE CNF

8 **Design, and implementation of a Java execution environment**
Chen, F.G.; Ting-Wei Hou;
Parallel and Distributed Systems, 1998. Proceedings., 1998 International Conference on , 14-16 Dec. 1998
Pages:686 - 692

[Abstract] [PDF Full-Text (56 KB)] IEEE CNF